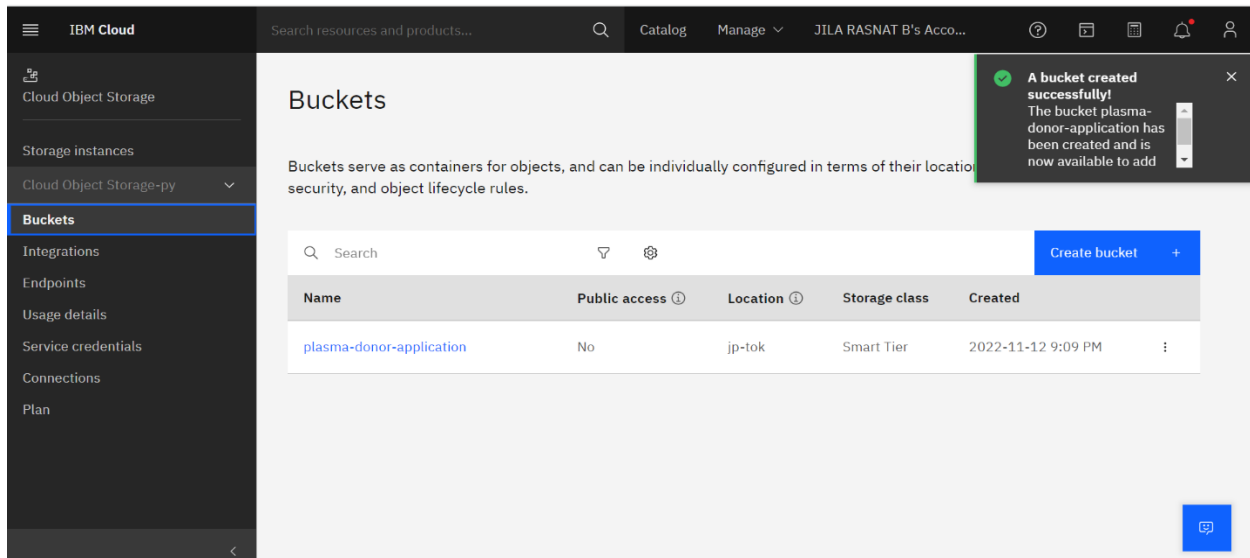


## Assignment 3

Assignment Date	02-11-2022
Student Name	EISHA V S
Student Roll Number	711719104028
Team ID	PNT2022TMID31589

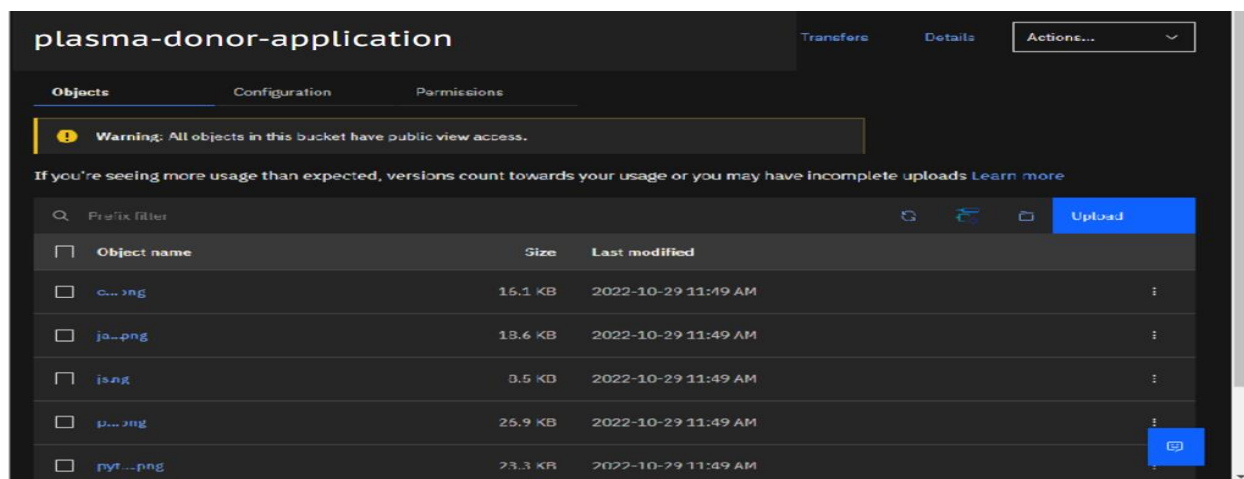
### 1. Create a Bucket in IBM object storage.



The screenshot shows the IBM Cloud Object Storage interface. On the left, a sidebar lists navigation options: Cloud Object Storage, Storage instances, Cloud Object Storage-py, Buckets (selected), Integrations, Endpoints, Usage details, Service credentials, Connections, and Plan. The main content area is titled 'Buckets' and includes a search bar, filters, and a 'Create bucket' button. A table lists the existing bucket 'plasma-donor-application' with details on public access, location, storage class, and creation time. A success notification in the top right corner states: 'A bucket created successfully! The bucket plasma-donor-application has been created and is now available to add'.

Name	Public access	Location	Storage class	Created
plasma-donor-application	No	jp-tok	Smart Tier	2022-11-12 9:09 PM

### 2. Upload any 5 images to IBM object storage and make it public. Write HTML code to display all the 5 images.



The screenshot displays the 'plasma-donor-application' bucket in IBM Cloud Object Storage. It shows a warning that all objects have public view access. Below the warning, a table lists five uploaded objects: 'c...png', 'ja...png', 'js.png', 'p...png', and 'pyt...png'. Each entry includes its size and the last modified date (2022-10-29 11:49 AM). An 'Upload' button is visible in the top right corner of the object list.

Object name	Size	Last modified
c...png	15.1 KB	2022-10-29 11:49 AM
ja...png	18.6 KB	2022-10-29 11:49 AM
js.png	0.5 KB	2022-10-29 11:49 AM
p...png	25.9 KB	2022-10-29 11:49 AM
pyt...png	23.3 KB	2022-10-29 11:49 AM

## HTML code :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Images in Cloud Object Storage</title>
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,400;1,200;1,300&display=swap" rel="stylesheet">
  <link rel="stylesheet" href="https://s3.jp-tok.cloud-object-storage.appdomain.cloud/ibm-assignment3-images-bucket/style.css"/>
</head>

<body>
  <nav>
    <h3>Images from Cloud Object Storage</h3>
  </nav>
  <div class="main-div">
    <div>
      
      <h3>Sachin Tendulkar</h3>
    </div>

    <div>
      
      <h3>Mahendra Singh Dhoni</h3>
    </div>

    <div>
      
      <h3>Rohit Sharma</h3>
    </div>

    <div>
      
      <h3>Virat Kohli</h3>
    </div>
  </div>
</body>
</html>
```

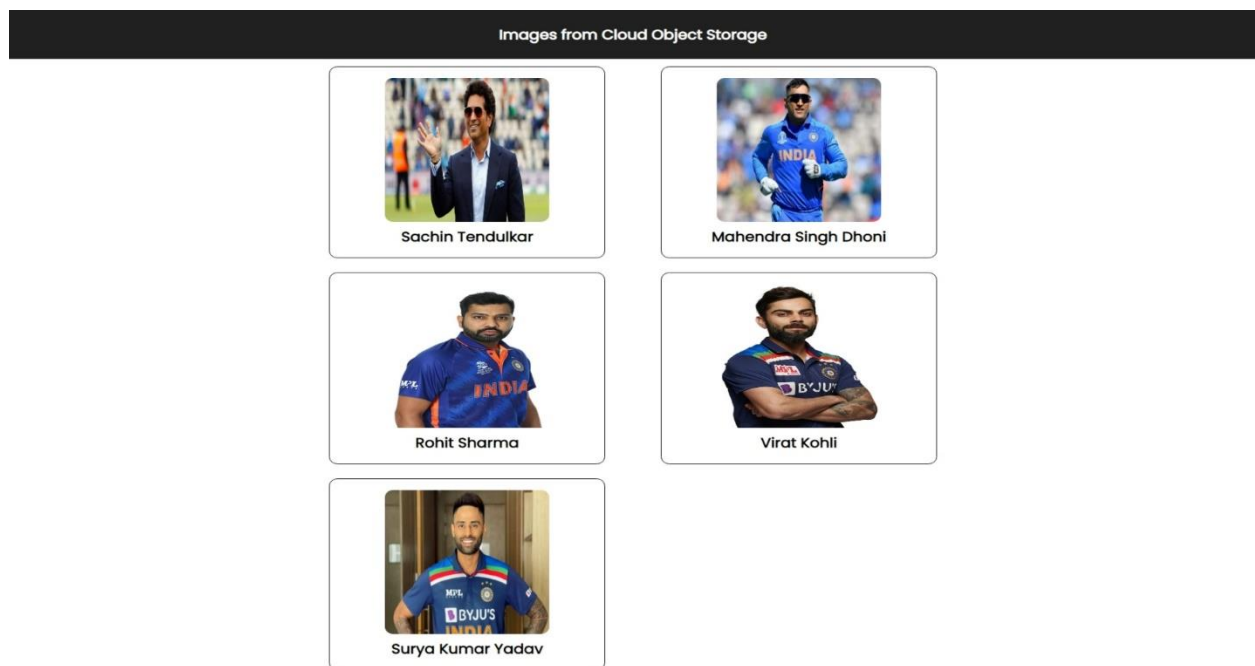
```

    </div>

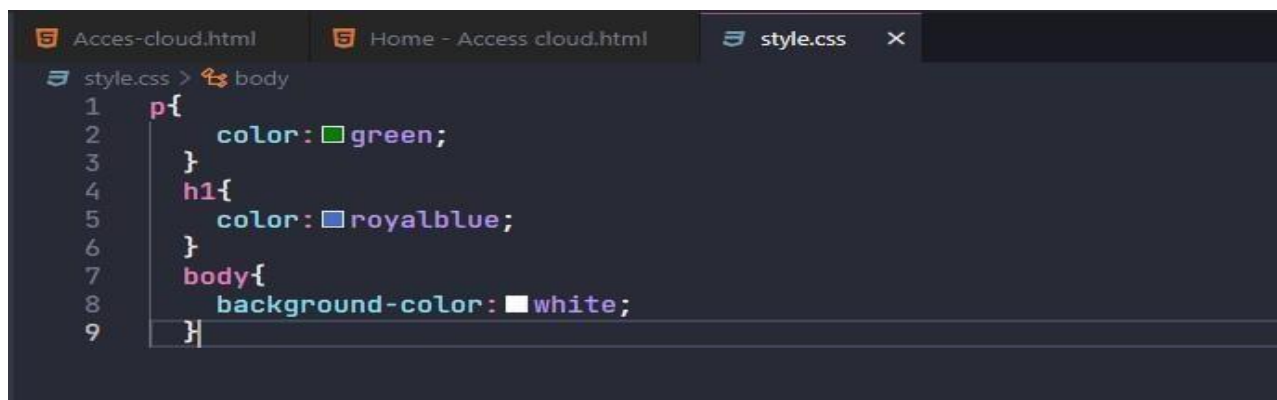
    <div>
        
        <h3>Surya Kumar Yadav</h3>
    </div>
</div>
</body>
</html>

```

## OUTPUT :



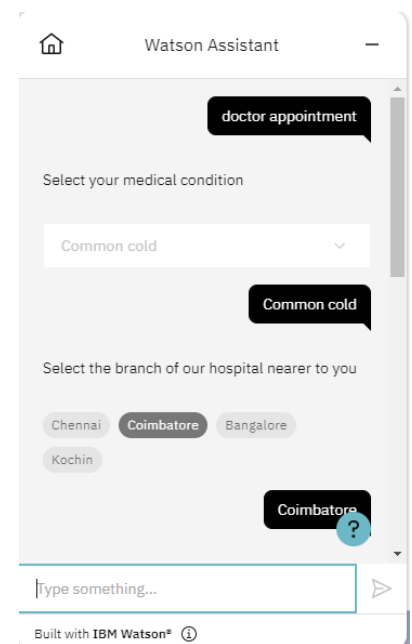
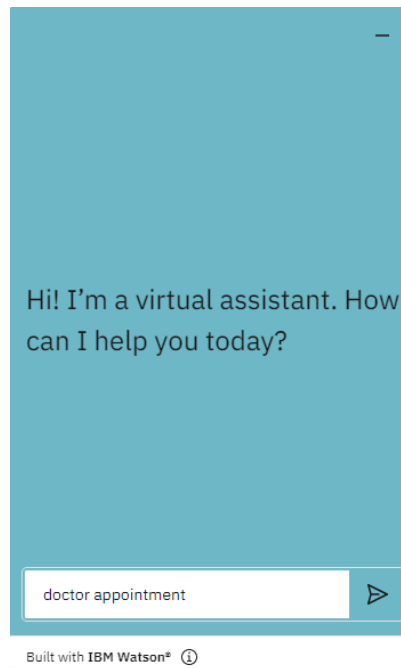
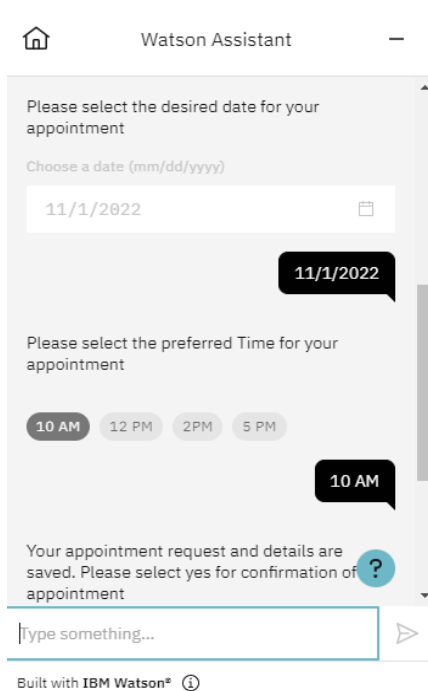
3. Upload a CSS page to the object storage and use the same page in your HTML code.

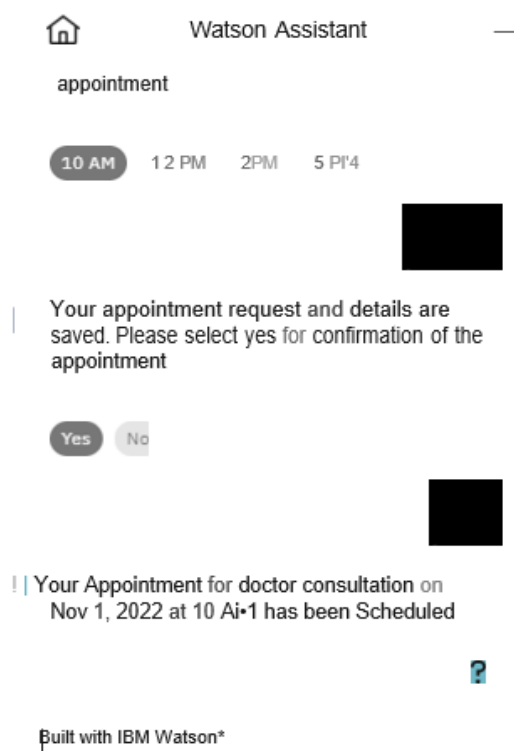


## WELCOME

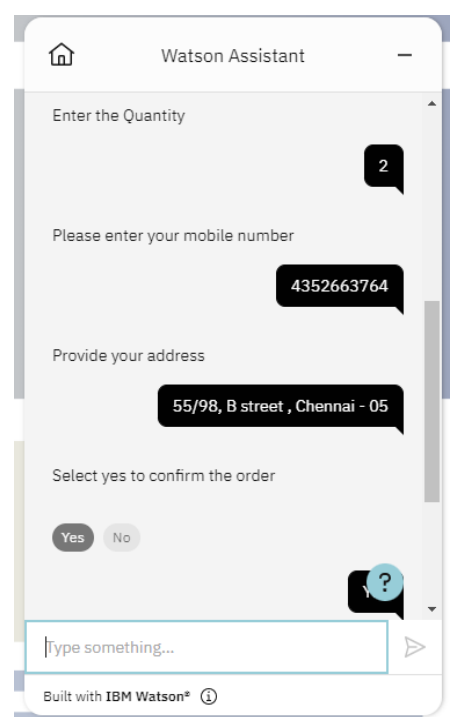
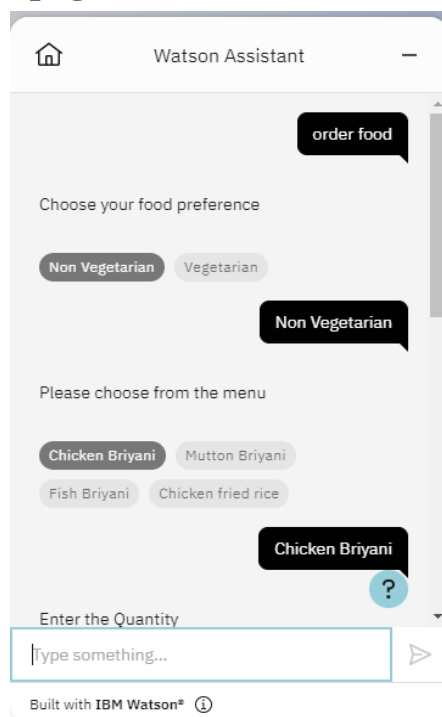
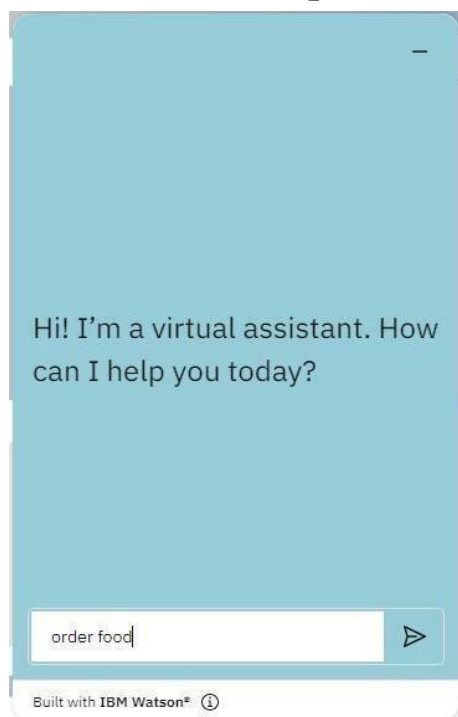
Click here to access the CSS file present in IBM cloud storage [Click here](#)

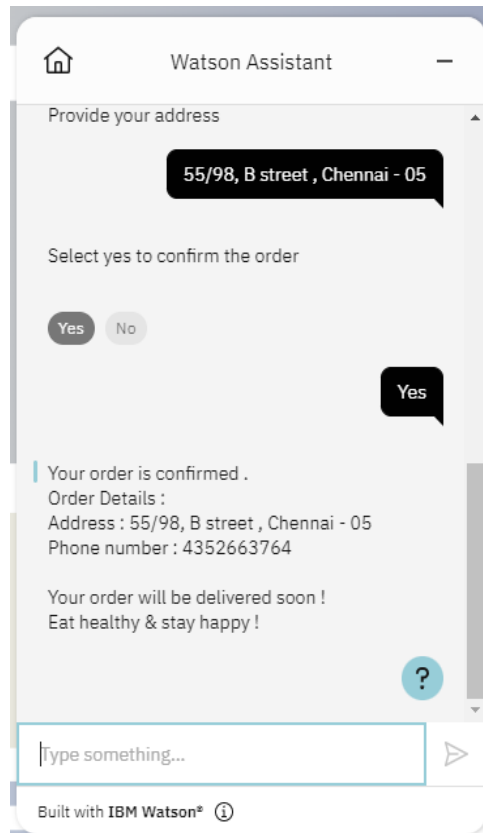
- 4. Design a chatbot using IBM Watson assistant for hospital. Ex: User comes with query to know the branches for that hospital in your city. Submit the web URL of that chat bot as a assignment.**





**5. Create Watson assistant service with 10 steps and use 3 conditions in it. Load that script in HTML page.**





## Script in HTML page:

```

1  <html>
2    <head>
3      <title>
4        SWIGGY BOT
5      </title>
6    </head>
7    <body>
8      <script>
9        window.watsonAssistantChatOptions = {
10          integrationID: "e80f8d5a-b636-4dce-be4e-10635f0d6014", // The ID of this integration.
11          region: "au-syd", // The region your integration is hosted in.
12          serviceInstanceID: "93b30170-ca16-4ef8-87ad-d5ae81cff433", // The ID of your service instance.
13          onLoad: function(instance) { instance.render(); }
14        };
15        setTimeout(function(){
16          const t=document.createElement('script');
17          t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
18            (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
19          document.head.appendChild(t);
20        });
21      </script>
22    </body>
23  </html>
24
25

```